Comparing Decoding Strategies for Subword-based Keyword Spotting in Low-Resourced Languages

William Hartmann\textsuperscript{1}, Viet-Bac Le\textsuperscript{2}, Abdel Messaoudi\textsuperscript{2}, Lori Lamel\textsuperscript{1}, Jean-Luc Gauvain\textsuperscript{1}

\textsuperscript{1}CNRS/LIMSI, Spoken Language Processing Group, 91403 Orsay cedex, France
\textsuperscript{2}Vocapia Research, 28 rue Jean Rostand, 91400 Orsay, France

\{hartmann, lamel, gauvain\}@limsi.fr, \{levb, abdel\}@vocapia.com

Abstract

For languages with limited training resources, out-of-vocabulary (OOV) words are a significant problem, both for transcription and keyword spotting. This paper investigates the use of subword lexical units for keyword spotting. Three strategies for using the sub-word units are explored: 1) converting word-based lattices to subword lattices after decoding, 2) performing a separate decoding for each subword type, and 3) a single decoding using all possible subword units. In these experiments, the best performance is achieved by carrying out a separate decoding for each subword type. Further gains are attained through system combination. We also find that ignoring word boundaries improves the detection of OOV keywords without significantly impacting in-vocabulary keyword detection. Results are presented on four languages from the IARPA Babel Program (Haitian Creole, Assamese, Bengali, and Zulu).

Index Terms: keyword search, spoken term detection, OOV, sub-word lexical units, low resource LVCSR

1. Introduction

Keyword spotting (KWS)—or spoken term detection (STD)—searches audio for a specific keyword—keywords can also be sequences of words. The task differs from the traditional speech-to-text (STT) task in that only a subset of words are important. For STT, every spoken word is equally important for the word error rate (WER) metric.

Out-of-vocabulary (OOV) keywords are a significant problem for KWS. While the OOV rate may be relatively low for a language, OOV words can represent a disproportionate number of keywords. Many methods have been previously proposed for detecting OOV keywords. A typical approach is to convert word lattices to phone lattices [1, 2]. Keywords are then detected by searching for a matching phone string. Converting the lattice to subwords is another alternative [3, 4]. Instead of converting the word lattice, additional hypotheses can be considered by using a phonemic confusion matrix [5]. Chen et al. take a similar approach, but from the keyword perspective [6]. When a word is not in the vocabulary, the search is expanded to include phonetically similar words in the original vocabulary.

These approaches all attempt to recover OOV words after decoding has already been performed. The OOV terms can also be anticipated by decoding with lexical units that are more likely to detect OOV terms. Previous work has used both phones as lexical units [7] and character ngrams [8]. Different types of subword units can also be combined and detected in a single decoding step [9].

In this work, we focus on the subword-based approach to KWS. Our goal is to examine three methods for using the sub-word units in the detection process: 1) converting word lattices to subword lattices, 2) using a separate decoding pass for each type of subword unit, and 3) combining all subword units together and performing a single decoding. Since the final method produces a lattice containing all possible subword units, we also simulate a subword unit-dependent decoding by only allowing a subset of subword units when searching for keywords.

Section 2 introduces the IARPA Babel data. Section 3 introduces the keyword spotting system. A detailed description of the decoding strategies is presented in Section 4. Section 5 presents the results and comparisons, and the concluding remarks are in Section 6.

2. IARPA Babel Data

2.1. Data Description

All experiments in this work use data from the IARPA-funded Babel program. The languages considered are Haitian Creole (iarpa-babel201b-v0.2b), Assamese (iarpa-babel_op1_103), Bengali (iarpa-babel_op1_102), and Zulu (iarpa-babel2006b-v0.1d). For each language, only the 10-hour subset of transcribed training data (limited language pack condition) is used for training acoustic and language models. Results are reported on a separate 10-hour development set.

For the keyword spotting experiments, we use the official evaluation keywords plus an additional set of development keywords. Table 1 provides an analysis of the keywords for each language. Note that a large number of keywords are not found in the development data and have no effect on the final results.

![Table 1: Distribution of keywords for each of the languages.](image)

<table>
<thead>
<tr>
<th>Language</th>
<th>IV Keywords</th>
<th>OOV Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creole</td>
<td>in Dev 2644</td>
<td>not in Dev 1055</td>
</tr>
<tr>
<td>Assamese</td>
<td>in Dev 2402</td>
<td>not in Dev 1439</td>
</tr>
<tr>
<td>Bengali</td>
<td>in Dev 2442</td>
<td>not in Dev 1300</td>
</tr>
<tr>
<td>Zulu</td>
<td>in Dev 1740</td>
<td>not in Dev 1196</td>
</tr>
</tbody>
</table>

Zulu is the most challenging of the four languages. In addition to the complex morphology, it is tonal and has clicks as part of its phonology. Both Assamese and Bengali share a non-
Roman script—with minor variations. Haitian Creole is similar to French and has a limited morphology.

2.2. Performance Metric

Maximum term-weighted value (MTWV) and actual term-weighted value (ATWV) are defined as the measures of interest for IARPA Babel program. ATWV was also used in the NIST 2006 Spoken Term Detection evaluation [10]. The keyword specific ATWV for keyword \( k \) at a specific threshold \( t \) can be computed by

\[
\text{ATWV}(k, t) = 1 - P_{FR}(k, t) - \beta P_{FA}(k, t)
\]

where \( P_{FR} \) and \( P_{FA} \) refer to the probability of a false reject (miss) and false accept, respectively. The constant \( \beta \)—set to a value of 999.9—mediates the trade off between false accepts and false rejections. MTWV represents the maximum score that could be obtained by using the optimal value for \( t \) over all keywords. We report results on MTWV in this work, but the differences between ATWV and MTWV are small. The range for MTWV (and ATWV) is between \(-\infty\) and 1; incorrect hypotheses are worse than no hypotheses.

In this performance metric, every keyword is equally weighted regardless of its frequency. Missing a single occurrence of a rare word can affect the final score as much as missing a more common word dozens of times. This explains why so much effort is devoted to the detection of OOV keywords. Wegmann et al. have a more detailed discussion of ATWV and MTWV in relation to the IARPA Babel program [11].

3. Keyword Spotting System

3.1. Acoustic Model Training

The LIMSI STK toolkit [12] was used to train and decode all systems. All voice activity detection (VAD) was performed by the BBN VAD system [13]. The BUT stacked bottleneck features [14] are also used. More details about the acoustic models can be seen in [15]. Language models are built using only the 10 hours of transcribed training data. A bigram model is used to generate the initial lattice, but a 3-gram language model is used when converting to the consensus network.

In order to remove the necessity of a pronunciation lexicon and to easily generate pronunciations for subword units, we use graphemes as our acoustic units. In addition, we use position-independent acoustic units so that the same acoustic model can be used for all subword types. The number of graphemes for each language are: Haitian Creole, 31; Assamese, 49; Bengali, 48; and Zulu, 26. Each system also uses three additional acoustic units for non-speech (silence, breath, and filler words).

WER is quite high for the Babel data [16], even using state-of-the-art ASR systems and techniques. This is not unexpected as the data consists of conversational telephone speech with limited training data. When using the acoustic models described in this paper for transcription, the WER ranges from 51.5% for Creole to 65.2% for Zulu [15].

3.2. Keyword Spotting

Using a bigram language model, a single pass decoding generates a lattice. The lattice is rescored with a trigram language model and converted to a consensus network (CN) [17] prior to the keyword search. Our preliminary experiments found better OOV performance using the CN compared to the lattice; the CN contains sequences of words that do not exist in the original lattice. Since many of the keywords are actually sequences of words, we search the CN for any matching sequence of words, up to 500ms of silence are permitted between any two words.

In order to detect more OOV keywords, we also ignore word boundaries when searching for matches. As per the guidelines of the task, keyword matches are case-insensitive. All experiments follow the no test audio re-use (NTAR) condition; all audio is processed before the keywords are known [18].

The CN contains a posterior value for each hypothesis. For multi-word keywords, we use the geometric mean of the individual posteriors. Before evaluating the results, we apply the BBN score normalization procedure [19]. Score normalization significantly improves results and brings the ATWV results near the MTWV results.

3.3. Subword Units

We use two main types of subword units. Note that we also explored using single character graphemes (a common approach in the literature [1]), but preliminary results on Bengali were poor, so we did not explore it further. The first type of units are created by Morfessor [20], a tool for morphological decomposition. Given a set of words and their frequency, Morfessor learns a generative model that uniquely decomposes any word into a sequence of morphological units.

The second type of subword unit is based on character ngrams. To generate these units, two properties are first defined—the maximum length of any subword unit and whether cross-word subword units are considered. Given those two properties, a set of all possible subword units are constructed and used to build a uniform language model. The training corpus is segmented using the uniform language model. As all words are equally probable, this is equivalent to minimizing the total number of units in the segmentation. A new trigram language model is built from the segmented training corpus, and the corpus is regenerted. This process is repeated until convergence. Unlike the previous subword type the segmentation for any word is not necessarily unique; the segmentation depends on the surrounding context.

This work considers character ngrams of length 3, 5, and 7. The word-internal subword units are referred to as 3gram-wi, 5gram-wi, and 7gram-wi throughout the remainder of the paper. Cross-word subword units are referred to as 3gram-cw, 5gram-cw, and 7gram-cw. These subword units were also used in a previous study that demonstrated the efficacy of cross-word subword units on Turkish data [21].

4. OOV Keyword Detection Approaches

4.1. Lattice Mapping

One approach to OOV keyword detection is to perform recognition with the standard word units and then map the lattice to subword units. To test this approach, we use both types of subword units described in Section 3.3. For the subword units based on character ngrams, we only consider the word-internal versions. As mentioned previously, the segmentation may not be unique for each word, so we only use the most likely segmentation. After the words in the lattice have been converted to the subword units, the lattice is converted to a CN. One of the major benefits of this approach is that many types of units can be examined with a single decoding of the data. However, there is a computational drawback. If the subword units are short, the final CN will contain large numbers of short units. Given a large number
of keywords, particularly if they are long, it can take significant
time to search the CN for the keywords.

### 4.2. Subword Unit-Dependent Decoding

An alternative approach is to perform a separate decoding for each
type of subword unit. After the training corpus has been segment-
a, a language model is built using the subword units. Ass-
ning the same training process, a language model based on
subword units obviously contains less context—negatively im-
pat-ting WER—but it allows for the recognition of words not
seen in training. There are approaches to compensate for this
effect [22], but they are not explored in this work. Based on this
language model, a lattice is decoded and converted to a CN.

The drawback to this approach is the computational cost.
The additional cost is linear in the number of subword unit
types. However, since the acoustic units are position-
indep-end, increasing the types of subword units does not af-
tect training time.

### 4.3. Joint Subword Unit Decoding

We can reduce the computational cost of using subword units by
combining all subword unit types into a single system. Given
multiple segmentations of the training transcript—one for each
subword unit type—a single language model is built over all
segmentations. We test two variants of this approach. The first
only includes subword units, while the second also allows the
inclusion of the original word units.

We also attempt to simulate subword unit-dependent de-
coding with the joint subword systems. Given a set of subword
units, all other units in the CN are removed; the additional prob-
ability mass is distributed among the remaining hypotheses in
proportion to their original confidence score. While this gives
the ability to only consider certain types of units during search,
it is not equivalent to unit-dependent decoding.

### 5. Results

#### 5.1. Word Boundaries

Before comparing the performance of the subword units and
decoding strategies, we show baseline performance and investi-
gate the effects of word boundaries. Table 2 lists two results per
language. All results use words as lexical units. The difference
is whether word boundaries are considered. When word bound-
aries are ignored, a significant portion of the OOV keywords are
detected—with little effect on IV keywords. The pattern for
subword units is similar, though the gain for OOV keywords is
smaller. Based on these results, the remaining experiments will

#### Table 2: MTWV results comparing the effects of ignoring word
boundaries when detecting keywords. All results use words as
lexical units.

<table>
<thead>
<tr>
<th>Language</th>
<th>Word Boundary</th>
<th>All</th>
<th>IV</th>
<th>OOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creole</td>
<td>keep</td>
<td>0.4510</td>
<td>0.4933</td>
<td>0.0000</td>
</tr>
<tr>
<td>Creole</td>
<td>remove</td>
<td>0.4554</td>
<td>0.4996</td>
<td>0.1518</td>
</tr>
<tr>
<td>Assamese</td>
<td>keep</td>
<td>0.2667</td>
<td>0.3425</td>
<td>0.0000</td>
</tr>
<tr>
<td>Assamese</td>
<td>remove</td>
<td>0.2884</td>
<td>0.3496</td>
<td>0.0737</td>
</tr>
<tr>
<td>Bengali</td>
<td>keep</td>
<td>0.2440</td>
<td>0.3200</td>
<td>0.0000</td>
</tr>
<tr>
<td>Bengali</td>
<td>remove</td>
<td>0.2741</td>
<td>0.3288</td>
<td>0.1014</td>
</tr>
<tr>
<td>Zulu</td>
<td>keep</td>
<td>0.1906</td>
<td>0.3523</td>
<td>0.0000</td>
</tr>
<tr>
<td>Zulu</td>
<td>remove</td>
<td>0.2280</td>
<td>0.3545</td>
<td>0.0806</td>
</tr>
</tbody>
</table>

#### Table 3: MTWV results on OOV keywords. Conversion refers
to converting words in the lattice to subword units prior to
searching. Dependent refers to using a separate decoding per
subword unit type. Simulated refers to performing decoding
with all subword units, but limiting the search to a specific type
(see Section 4.3). Note that we do not convert words to cross-
word subword units, so no results exist for those cases.

<table>
<thead>
<tr>
<th>Language</th>
<th>Lex. Unit</th>
<th>Conversion</th>
<th>Dependent</th>
<th>Simulated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creole</td>
<td>morфессor</td>
<td>0.2076</td>
<td>0.2994</td>
<td>0.2257</td>
</tr>
<tr>
<td>Creole</td>
<td>3gram-wi</td>
<td>0.2139</td>
<td>0.3349</td>
<td>0.2813</td>
</tr>
<tr>
<td>Creole</td>
<td>3gram-cw</td>
<td>0.3439</td>
<td>0.2763</td>
<td>0.2635</td>
</tr>
<tr>
<td>Creole</td>
<td>5gram-wi</td>
<td>0.2181</td>
<td>0.3361</td>
<td>0.2815</td>
</tr>
<tr>
<td>Creole</td>
<td>5gram-cw</td>
<td>0.3484</td>
<td>0.3196</td>
<td>0.3076</td>
</tr>
<tr>
<td>Creole</td>
<td>7gram-wi</td>
<td>0.2223</td>
<td>0.3630</td>
<td>0.3030</td>
</tr>
<tr>
<td>Creole</td>
<td>7gram-cw</td>
<td>0.3890</td>
<td>0.3229</td>
<td></td>
</tr>
<tr>
<td>Assamese</td>
<td>morфессor</td>
<td>0.0765</td>
<td>0.1280</td>
<td>0.0978</td>
</tr>
<tr>
<td>Assamese</td>
<td>3gram-wi</td>
<td>0.0899</td>
<td>0.1302</td>
<td>0.0941</td>
</tr>
<tr>
<td>Assamese</td>
<td>3gram-cw</td>
<td></td>
<td>0.1430</td>
<td>0.0996</td>
</tr>
<tr>
<td>Assamese</td>
<td>5gram-wi</td>
<td>0.0842</td>
<td>0.1648</td>
<td>0.1139</td>
</tr>
<tr>
<td>Assamese</td>
<td>5gram-cw</td>
<td></td>
<td>0.1348</td>
<td>0.1143</td>
</tr>
<tr>
<td>Assamese</td>
<td>7gram-wi</td>
<td>0.0898</td>
<td>0.1669</td>
<td>0.1169</td>
</tr>
<tr>
<td>Assamese</td>
<td>7gram-cw</td>
<td></td>
<td>0.1562</td>
<td>0.1123</td>
</tr>
<tr>
<td>Bengali</td>
<td>morфессor</td>
<td>0.0918</td>
<td>0.1622</td>
<td>0.1034</td>
</tr>
<tr>
<td>Bengali</td>
<td>3gram-wi</td>
<td>0.1184</td>
<td>0.1774</td>
<td>0.1151</td>
</tr>
<tr>
<td>Bengali</td>
<td>3gram-cw</td>
<td></td>
<td>0.1433</td>
<td>0.1159</td>
</tr>
<tr>
<td>Bengali</td>
<td>5gram-wi</td>
<td>0.1147</td>
<td>0.1819</td>
<td>0.1328</td>
</tr>
<tr>
<td>Bengali</td>
<td>5gram-cw</td>
<td></td>
<td>0.1664</td>
<td>0.1329</td>
</tr>
<tr>
<td>Bengali</td>
<td>7gram-wi</td>
<td>0.1145</td>
<td>0.1792</td>
<td>0.1328</td>
</tr>
<tr>
<td>Bengali</td>
<td>7gram-cw</td>
<td></td>
<td>0.1646</td>
<td>0.1326</td>
</tr>
<tr>
<td>Zulu</td>
<td>morфессor</td>
<td>0.0845</td>
<td>0.2539</td>
<td>0.2114</td>
</tr>
<tr>
<td>Zulu</td>
<td>3gram-wi</td>
<td>0.1026</td>
<td>0.2904</td>
<td>0.1772</td>
</tr>
<tr>
<td>Zulu</td>
<td>3gram-cw</td>
<td></td>
<td>0.2736</td>
<td>0.1926</td>
</tr>
<tr>
<td>Zulu</td>
<td>5gram-wi</td>
<td>0.0666</td>
<td>0.2817</td>
<td>0.2581</td>
</tr>
<tr>
<td>Zulu</td>
<td>5gram-cw</td>
<td></td>
<td>0.2846</td>
<td>0.2669</td>
</tr>
<tr>
<td>Zulu</td>
<td>7gram-wi</td>
<td>0.0661</td>
<td>0.2909</td>
<td>0.2652</td>
</tr>
<tr>
<td>Zulu</td>
<td>7gram-cw</td>
<td></td>
<td>0.2931</td>
<td>0.2658</td>
</tr>
</tbody>
</table>

#### 5.2. Lattice Conversion vs. Unit-Dependent Decoding

Table 3 compares the performance of several types of subword
units when converting the lattice or performing unit-dependent
decoding. Due to space constraints, only performance on OOV
keywords are shown. No subword unit gives better performance
than baseline word decoding for IV keywords. As expected,
performing a separate decoding for each type of subword unit
gives significantly better performance—ranging from 50% to
over 100% relative improvement. It is interesting to note the
subword-based units do not perform as well for Assamese and
Bengali, possibly due to their larger character and grapheme set.

In all cases, the worst performing subword type is the
Morfessor-based subword type, but this is misleading. The
Morfessor-based units consistently give the best IV keyword
performance—though still not as good as the baseline word
system—and can give the best overall performance depending
on the ratio of IV to OOV keywords. When comparing the char-
acter ngram-based subword unit results across languages, there
does not appear to be a pattern. The worst system for Creole
(5gram-wi) performs well for both Bengali and Zulu.

In a previous study on Turkish [21], we noted that cross-
word subword units did not outperform word-internal units, but
Table 4: MTWV results for the joint decoding systems. Joint hypotheses are averaged. More sophisticated combination methods could produce better results, but this provides a guide to the complementarity of the results. Table 5 contains the results for combining the results for each decoding strategy discussed in Section 4.

The lattice conversion results are at a slight disadvantage because the combination does not include the cross-word subword units, but it is clearly the worst performing approach. The simulated results offer better performance than the lattice conversion, but fail to obtain the performance of the joint system used to generate them. The best performance is from combining the subword unit-dependent results.

6. Conclusion

A major difficulty in the KWS task is the detection of OOV keywords. Subword-based lexical units are one potential solution. We have investigated three approaches to incorporating subword units into the decoding process. The simplest, converting a word lattice to subword units, also performs the worst. While it can detect some OOV keywords, the sequences of subword units that can form OOV words do not appear often.

The second approach, subword unit-dependent decoding, provides the best OOV performance, especially when combining multiple results. However, it does come at a higher computational cost since multiple decoding passes are required. Also, it is not clear if determining which subword unit types will perform best prior to decoding is possible.

The final approach, joint subword decoding, has worse IV performance than the baseline and worse OOV performance than the best subword unit-dependent result. However, depending on the percentage of OOV keywords, it can produce the best overall result. If a single system is required, it is a good candidate. We also explored the idea of simulating the unit-dependent systems using the joint results. The simulated performance is not as good, but provides a method for creating a large set of results that can later be combined.

We find the different subword unit results contain complementary information. Even using a simple combination approach, large gains in MTWV are seen. We plan to explore more sophisticated combination methods to improve results even further. The Bengali and Assamese gains were not as large as with Creole and Zulu. In future work we will investigate whether this is due to a specific characteristic of these languages, and whether an improvement to the subword unit-based methods can compensate for it.

7. Acknowledgements

We would like to thank other partners of the BABELON team on the IARPA-Babel project for exchanging the resources (especially BUT for the bottle-neck features, BBN for VAD, autotranscription, and score normalization tool). We would also like to acknowledge the help and contribution of other colleagues in LIMSI and Vocabia.

This research was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.
8. References


